Development of a Robot Cell with a Vision System

E. LIEBENBERG, I.A. GORLACH
Department of Mechatronics
Nelson Mandela Metropolitan University
Gardham Avenue, North Campus, Summerstrand, Port Elizabeth
SOUTH AFRICA
s210212225@live.nmmu.ac.za, igor.gorlach@mandela.ac.za, http://www.nmmu.ac.za

Abstract: - Modern manufacturing systems employ robots for industrial processes and material handling tasks. Robotic systems facilitate reconfigurability and flexibility, which provide rapid manufacturing of customised products of high quality to meet the worldwide market demands. The application of Artificial Intelligence (AI) together with robotics brings many benefits and enhancements of production processes. The intelligence of a robotic system is facilitated by advanced sensors, of which a computer vision is the most important. In this paper, the development of a robotic cell with a Fanuc M-1iA parallel kinematics robot, a conveyor and a vision system is presented. The aim is to design a flexible robotic system for picking and sorting of parts, which are supplied randomly by a continuous conveyor. The developed system provides a low-cost solution, which employs a flexible conveyor, an all-purpose USB camera for image capturing and Visual Studio for image processing. All the elements of the system were successfully integrated with the robot controller. A user-friendly human machine interface allows easy and fast camera calibration and interaction with the robotic cell environment. The main advantages of the developed robotic system are its low-cost and flexibility. The system can be used for the demonstration of capabilities of intelligent robotic systems and training purposes.

Key-Words: - automation, robots, vision system

1 Introduction

Intelligent robotic systems integrate various hardware and software platforms such as fixtures, conveyors, part feeding devices, sensors, cameras, controllers etc. Machine vision is the core of intelligent robot systems. Therefore, the application of machine vision in robotics has been a research area in engineering for decades [1] – [11]. Although physical imaging sensors, such as CCD, CMOS and 3-D lasers, are important, the main focus of research lies in the field of image acquisition and processing.

There are four main parameters that are important in machine vision: object dimensions, surface texture, structural quality and operational quality [1]. Most of robotic vision systems are used for inspection, handling and sorting operations of 2-D or 3-D parts. In many robotic applications, handled objects can be moving, which increases the complexity of the process, but improves productivity and flexibility. Hence, in this project, parts to be sorted are presented on a moving conveyor.

A digital image (bitmap) consists of an array of pixels, which locations are identified by their X and Y coordinates, measured from the top left-hand corner. Pixels have attributes such as the grey level or brightness and the RGB (red, green and blue) level. The colour values represent both the colour and the brightness or greyscale of the individual pixel. The greyscale image is represented by a histogram, which can be used for different applications by means of pixel filtering or intensification of the greyscale or a certain colour level, for example, for part detection and identification [5].

Image processing includes histogram analysis, thresholding, convolution mask, connectivity, edge/blob analysis and noise reduction [6]. Blob detection provides a big variety of filtering methods as blobs can be filtered based on size, colour location etc. The issue with blobs may arise when objects are overlapping. This can be resolved with edge detection by highlighting key contours and features in the field of view [7]. Various algorithms and methods developed for edge detection are available [3, 8]. The process of an edge detecting algorithm is composed of three stages, such as: smoothing, differentiation and labelling. Another big factor in image processing is noise reduction, which can be achieved with a number of techniques reported [9, 10]. Since the ultimate goal of this project is object sorting, it is important to achieve an efficient and reliable process of image processing. The technique applied in this research includes a number of sequential and parallel steps, such as noise reduction, image segmentation, dimension, colour and shape analyses, and matching and sorting decisions.

The developed intelligent robotic cell for picking and sorting parts consists of a Fanuc M-1iA lightweight parallel kinematic robot, a machine vision system and a flexible conveyor. The vision system identifies moving parts and determines their coordinates, which are then sent to the robot, which picks and sorts them according to their colour.

2 Development of the Robotic Cell

2.1 Camera Selection

The comparison of cameras was based on the costs and the following parameters: resolution, sensor sensitivity, frame rate, colour, interface and software. A number of options were considered in camera selection, including: a Logitech C270 HD webcam, a Blackfly PoE GigE Mono camera, a E0-0312M industrial monochrome camera and a ELP 2.0 MP USB camera. The latter was chosen for the application in the robotic cell. It is an all-purpose USB camera, which provides multiple adjustable settings including resolution and frame rate up to 120 Mb/s and it is typically used for non-vital applications such as security systems or automobile data recorders (dash-cams).

2.2 Image acquisition and processing

The aim of the project was to sort parts using a vision system with the colour as the means to differentiate between parts. Because the focus was on identifying parts based on colour, poker chips were selected to represent parts. The decision for using poker chips as parts was based on the fact that they are similar in size, shape and weight, but differ in colour. Furthermore, when poker chips are placed flat on the conveyor they appear flat and broad compared to their height when viewed at a slight angle. This reduces the size of the shadow cast by the object, simplifying the vision system's procedure to accurately distinguish between the object and its background for part identification and classification.

C# computer language was chosen for this project due to its ability (a) to connect with the Fanuc robot's controller through the Fanuc interface software and (b) directly interface with USB cameras allowing for image capture and analysis. To aid in image processing and analysis, the .NET framework and particularly the library AForge.NET was used. The AForge.NET framework is widely applied in the field of artificial intelligence, machine vision, machine learning, genetic algorithms and robotics. This framework supplies the user with libraries that allow multiple image processing procedures, including image acquisition, image filtering, and blob detection. The vision system program controls the following processes: the camera calibration procedure, setting of the colour filters and size constraints for filtering noise from parts, starting the conveyor and determining the robot coordinates for parts traveling on the conveyor. Due to the physical space constraints of the robotic cell, the camera was positioned at an angle as shown in Fig. 1, which resulted in errors of the model that would linearly relate the image coordinates to the corresponding robot coordinates. The initial approach was to create a transformation matrix that would relate the camera's reference frame to the robot's reference frame. Thereafter, a differential evolution (DE) program was created to solve the corresponding matrix variables. Although this method succeeded in calculating the matrix variables when viewed from above, the method was ultimately unsuccessful in accurately calculating the variables as the average error per point was approximately 1 mm.

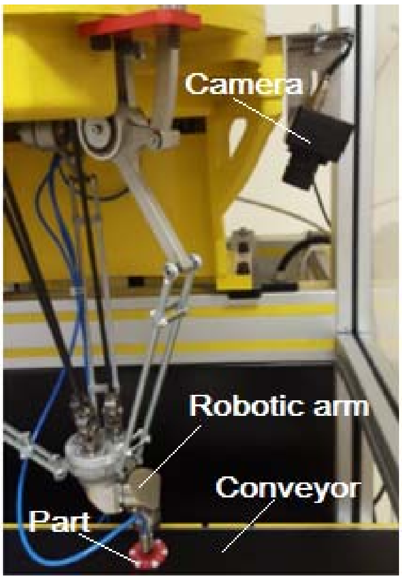

Fig. 1 Physical set-up of the robotic cell

This problem was overcome by using projective mapping, which is a tool of warping a source image to fit into a destination space. The procedure starts by selecting four corners from both the source and destination quadrilaterals. Thereafter, a transformation matrix is created that maps the points from the original quadrilateral to that of the destination image [11]. As shown in Fig. 2, a calibration white page with black dots is placed on the conveyor in the robot workspace and the camera view. The user is then required to select the four corners of the rectangle on the source image. These four corners serve as the four outer corners of all subsequent source images. All the pixels of following images that fall within the quadrilateral formed by these four corners are then projected onto a rectangular plane with predetermined dimensions. After the projective mapping procedure has been set up, the user can continue with the calibration process by linking the corresponding points between the camera's reference frame and the robot's reference frame to create a linear relationship between the two reference frames.

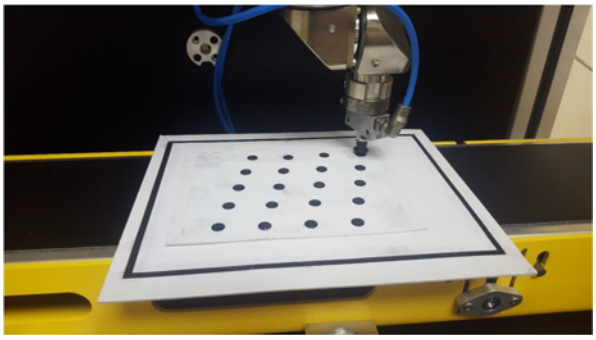

Fig 2. Calibration of the vision system

The robot's end effector with a vacuum suction cup is jogged to each corresponding point and the X and Y coordinates at each location are then saved. In order to determine the equation that defines the relationship between the two frames, linear regression is used. By detecting the coordinates of points of an image, to which the projective mapping process was applied, allows for change of depth to be accounted for, thereby reducing the resultant distortion and increasing the resultant accuracy. The image coordinates of objects arriving on the conveyor can then be used as input values to the equation determined by the linear regression procedure in order to determine the corresponding robot coordinates.

In order to distinguish between the parts on the conveyor and the background formed by the conveyor belt, a blob detection algorithm is used. The algorithm supplied by the AForge.NET libraries takes minimum and maximum object diameters as inputs before it is applied to a selected image. Furthermore, the algorithm treats black pixels as background and coloured pixels as objects. The vision system, therefore, allows the user to filter out a given amount of red, green and blue from the image. As a result of this filtering process, the darker pixels will turn black first, allowing the program to interpret them as background, whereas the brighter pixels will remain to form the objects.

2.3 Graphical User Interface

The vision system contains a graphical user interface (GUI) that serves as the user's control interface, which allows the user control over the system's functions whilst providing feedback from the vision system. The developed GUI consists of three tabs. The first tab is the control tab, which permits the user to control the operation of the vision system. The control tab provides for the camera selection as well as the camera resolution selection. After making the required selection, the user can change the minimum and maximum object size by using the size sliders. The sliders also allow the user to control the amount of RGB filtered out by the colour filter. In addition to the filter settings, the control tab hosts the image feedback from the camera and the connect and start buttons, which connect the robot controller and start the conveyor belt, respectively. Lastly, the coordinates of the detected blobs are displayed in a window as shown in Fig. 3.

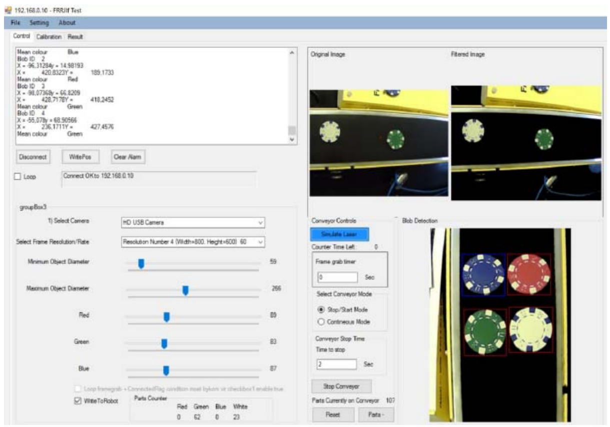

Fig 3. Graphical User Interface

The second and third GUI tabs are used to calibrate the vision system. The second tab contains the calibration controls and the third tab contains an 800x600 size image after converting the colour image to greyscale, and subsequently applying a threshold value to the greyscale image to further convert it into a binary image.

The controls on the second GUI tab allow the user to take a single photo with the camera, after which the conversion to a binary image takes place with the threshold value set by the user. The image captured is displayed on the third GUI tab called Result tab. After an image has been taken the result tab allows the user to set the four corners required for the image warping procedure, compensating for the image's changing depth. If the four corners of the calibration image are selected, a warped image will appear on the second tab containing a section of the original image.

A blob detection algorithm similar to the one used during normal operation is applied to an inverted image of the warped image displayed on the second tab. This allows a white page with black to be used as a calibration page whilst the blob detection program is applied to a black page containing white dots. Furthermore, the calibration tab contains a save blobs button to save the coordinates of the blobs detected as well as the current robot position for calibration purposes. If all the blob locations and their corresponding robot positions have been saved, then the user may calibrate the axes by selecting them from dropdown boxes. Lastly, the calibration tab provides a means to save and display the locations used to place and temporarily store parts during the sorting process, as well as the conveyor locations for each colour part when transferred back to the conveyor. These locations are labelled Shelf and Conveyor stacks, respectively.

2.4 System Integration

Fig. 4 shows the system architecture indicating how the various subsystems are connected to achieve an operational system. From the diagram, it can be seen that the robot controller is a host or a cell controller. In order for the vision system to communicate with the conveyor control it has to first relay the information to the robot controller. The same is true for the conveyor control program. A PIC18F6520 microcontroller is used for the conveyor control and it is connected to the robot controller via a Digital I/O.

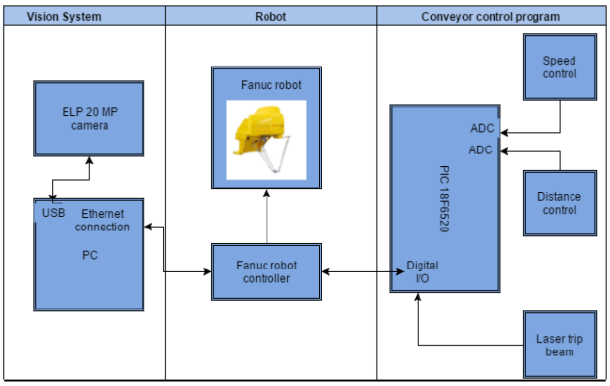

Fig. 4 System architecture

The vision system is connected with the robot's controller through an Ethernet connection between the PC and the robot controller. Libraries from the Fanuc interface were used to aid in establishing this connection. Once connected the vision system creates a data table that contains variables representing the various registers of the robot controller that are accessible through the Ethernet connection.

The control program of the robotic cell compiled in C# consists of two subprograms, namely A and B, which flow diagrams are shown in Fig. 5 and 6, respectively. By polling the data table's refresh function, the vision system can monitor certain robot input/outputs. This allows using the robot's digital I/O pins as flags in order to control the program's logic. Once a monitored flag has been set, the vision system can execute a predetermined series of instructions, after which it sets another flag in the data table indicating to the robot controller that the process has been completed. Similarly, the robot's controller constantly refreshes its data table, monitoring the digital IOs and position registers.

This process is used to indicate to the vision system when to process an image for part detection and determine the part's location, which is represented by the flow diagram of subprogram A shown in Fig. 5. A coordinate-required bit is used as a flag, and is constantly polled by refreshing the data table. After the bit is set high by the microcontroller as a result of the part travelling its predefined distance on the conveyor belt, the vision system processes the next available image in order to determine the object's coordinates.

When the image processing and blob detection processes are completed, the part's coordinates are uploaded as variables in the data table that represents the robot controller's position registers as shown in Fig. 6. A coordinate-available bit is set in the data table indicating to the robot controller program that there are coordinates available in the data table. When the robot program detects that the coordinates-available bit is set high it allows the end effector to move to the locations stored in the position register, thereby moving to the part on the conveyor to pick up the part.

3 Results and Analysis

A linear regression analysis was used to calibrate and evaluate the accuracy of the camera's reference frame in relation to the robot coordinate frame, using the methodology described in [12]. Regression analysis aims to produce an m*th* order polynomial of the following form:

$$y_c = a_0 + a_1x + a_2x^2 + .. + a_mx^m \qquad (1)$$

Where: a_0, a_1, a_2,...a_m are the coefficients of the polynomial, x is the independent variable, y_c is the corresponding dependent variable.

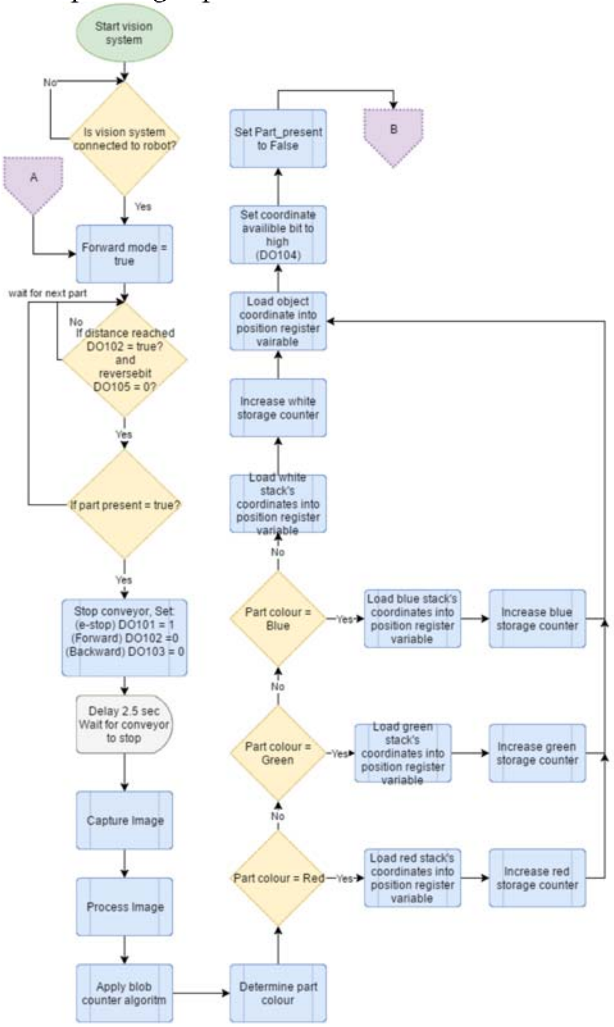

Fig 5. Flow diagram of subprogram A

The most common method of establishing a polynomial is the *least-squares* method, which aims to reduce the equation D, given below, to a minimum for a given order of polynomial:

$$D = \sum_{i=1}^{N} (y_i + y_{ci})^2 \tag{2}$$

Since the curve approximation may not pass through all the data points, there is a deviation, which can be characterised by the standard deviation of fit or an error as follows:

$$s_{xy} = \sqrt{\frac{\sum_{i=1}^{N} (y_i + y_{ci})^2}{v}} \tag{3}$$

Where: v is the degree of freedom of the fit and it is determined as follows:

$$v = N - (m + 1) \tag{4}$$

Where: N is the sample size.

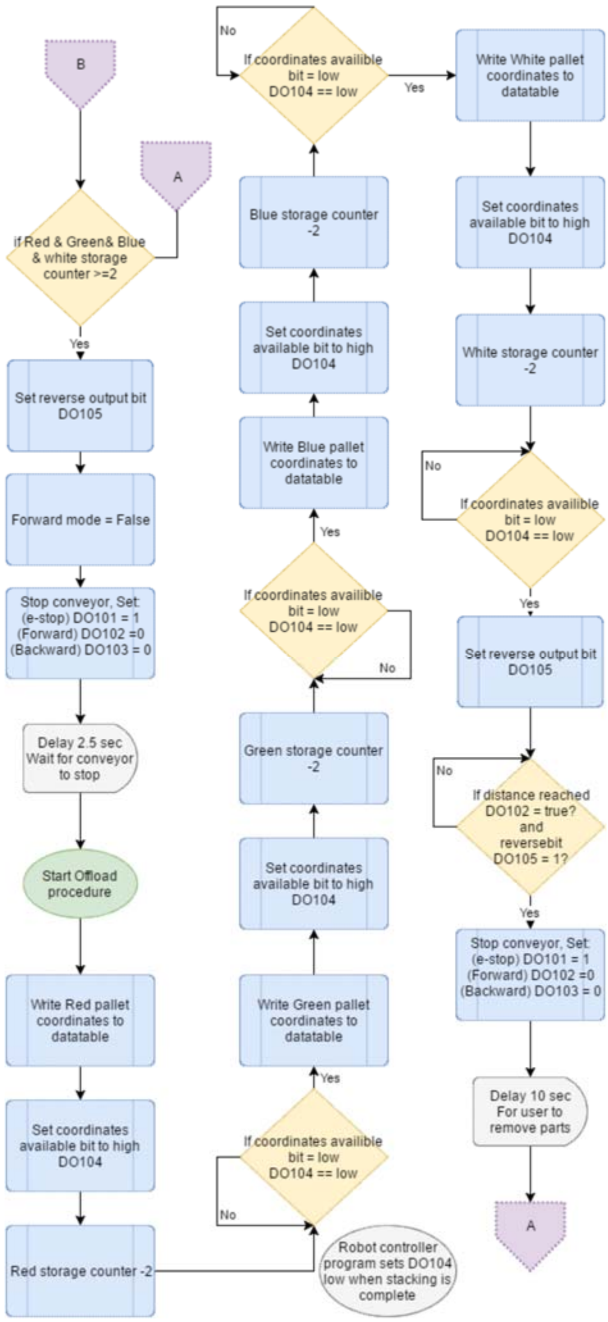

Fig 6. Flow diagram of subprogram B

Since it is probable that both the measured and the independent variable are likely to have some random error as a result of the measuring process, the confidence interval needs to be determined using a complete uncertainty analysis. The curve approximation resultant equation with the confidence level is expressed as follows [12]:

$$y_c \pm t_{v,P} \frac{s_{xy}}{\sqrt{N}} \tag{5}$$

Where: $t_{v,P}$ is the estimator for a coverage factor of the confidence level P, which is obtained from a distribution table. In order to calculate the respective 95% confidence intervals the estimator t value was selected for v = 18. This gives t=2.101. The confidence levels for the X and Y axes were determined as 0.152 mm and 0.335 mm, respectively. The linear approximation curves are shown in Fig. 7 and Fig. 8.

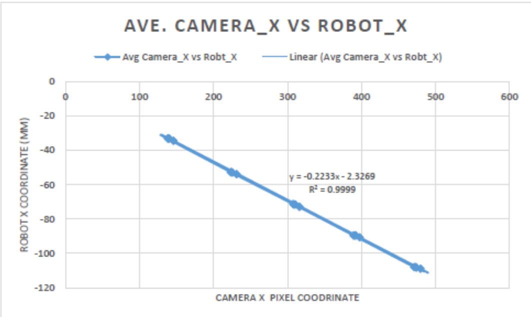

Fig. 7 Regression analysis results for the X axis

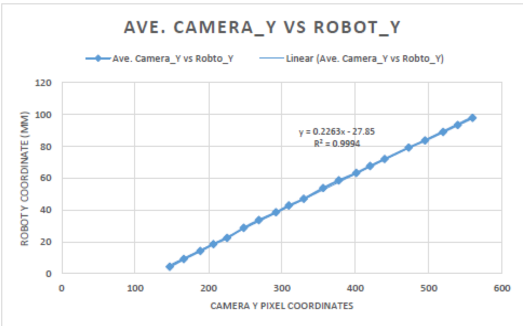

Fig. 8 Regression analysis results for the Y axis

4 Conclusion

In this research, an intelligent flexible robotic cell was developed consisting of a Fanuc M-1iA parallel kinematics robot, a conveyor and a vision system. The designed peripherals such as the conveyor, the vision system, the controller and software further enhanced the robotic platform, which could benefit teaching and simulation of production environment. The intelligent robotic system is capable of tracking the positions of parts entering the robot's working envelope, and pick, sort and place parts according to their colour. The project could challenge future students to further enhance the functionality of the robot or improve programs developed for this project.

This project proved that the vision and conveyor systems added to the Fanuc robot establish an improved level of automation and increased functionality of the robot. While C# was chosen as programming language, for future work, MatLab could also be considered as it may simplify the design process.

5 Acknowledgments

The authors would like to acknowledge the support and financial assistance provided by both General Motors South Africa through the GMSA Chair of Mechatronics.

References:

[1] E. N. Malamas, E. G. M. Petrakis and M. Zervakis, A Survey on Industrial Vision Systems Applications and Tools, [Online].

[2] S. Ng, J. M. Yusof, M. S. Kassim, and A. Rahman, "IV-FMC: An Automated Vision Based Part Modeling and Reconstruction System for Flexible Manufacturing Cells," IEEE International Conference on Robotics and Biomedics, pp. 1383-1387, 2009.

[3] R. Jain, R. Kasturi and B. G. Schunck, *Machine vision*, 1st ed. McGraw-Hill, 1995.

[4] M. Ren, J. Yang and H. Sun, Tracing boundary contours in a binary image. *Image and vision computing*, Volume 20, 2002, pp. 125-131.

[5] B. K. Horn, Determining Lightness from and Image, *Computer Graphics and Image Processing*, 1974, pp. 277-299.

[6] M. Sonka, V. Hlavac and R. Boyle, Image Processsing, Analysis, and Machine Vision. 2nd ed. Brooks/Cole Publishing Company, 1999.

[7] L. Zhou, G. Hua, D. Xu and H. Wu, Edge Detection Algorithm for Uneven Lighting Image based on Vision Theory, *International Conference on Computational Intelligence and Natural Computing*, Vol. 1, 2009, pp. 182-185.

[8] D. Ziou, and S. Tabbone, Edge Detection Techniques-An Overview, *Pattern Recognition and Image Analysis*, Vol. 8, 1998, pp. 537-559.

[9] B. Yeul-Min, C. Dong-Chan, L. Jin-Aeon and K. Whoi-Yul, Noise Reduction for Image Signal Processor in Digital Cameras, *International Conference on Convergence and Hybrid Information Technology*, 2008, pp. 474-481.

[10] Z. Y. Yao, Y. Wu and C. I. Wang, Research on Intelligent Sorting System Based on Machine Vision, *Second International Conference on Intelligent Computation Technology and Automation*, Vol. 463, 2009, p. 460.

[11] P.S. Heckbert, Carnegie Mellon University School of Computer Science, [Online].

[12] R. S. Figliola, and D. E. Beasley, *Theory and Design for Mechanical Measurements*, John Wiley & Sons Inc, 2011.